Exhibit 74-B

Redacted Version of Document Sought to be Sealed

GIBSON DUNN

Gibson, Dunn & Crutcher LLP

555 Mission Street
San Francisco, CA 94105-0921
Tel 415.393.8200
www.gibsondunn.com

Rosemarie T. Ring
Direct: +1 415.393.8247
Fax: +1 415.801.7358
RRing@gibsondunn.com

HIGHLY CONFIDENTIAL—ATTORNEYS EYES ONLY

April 18, 2022

VIA JAMS ACCESS

Special Master Daniel B. Garrie
DGarrie@jamsadr.com

Re: *In re Facebook Consumer Privacy User Profile Litigation*, JAMS Ref No. 1200058674

Dear Special Master Garrie,

Pursuant to the Special Master's Order Following March 9, 2022 Hearing Regarding Plaintiffs' Motion to Compel Production of Plaintiff Data, dated March 22, 2022, Facebook submits this proposal to assist in determining "what, if any" additional named plaintiff data should be produced, as directed in Judge Corley's order dated January 12, 2022 (Dkt. No. 807), and respond to "Scenario" questions.

Facebook's Proposal For Producing Additional Named Plaintiff Data

As Judge Corley explained in her most recent order on named plaintiff data, issued on January 12, 2022, the Special Master is working with the parties to determine "what, if any" additional named plaintiff data should be produced consistent with Rule 26. Facebook continues to believe that the named plaintiff data that has already been produced, which includes data in all three categories of "discoverable user data" identified in Judge Corley's Discovery Order No. 9, satisfies its obligations under Rule 26. The additional data Plaintiffs seek is neither relevant to their claims nor proportional to the needs of this case because it was not shared or made accessible to third parties.

That said, in an effort to resolve this issue, and consistent with Judge Corley's Discovery Order Nos. 11 and 12 allowing Plaintiffs to test Facebook's position on whether there is additional named plaintiff data that was shared or made accessible to third parties, Facebook proposes the protocol set forth below for producing additional named plaintiff data.

As explained in Facebook's March 7, 2022 submission to the Special Master, at a high level, user data is stored in production systems and the data warehouse. [redacted]. Accordingly, we propose to produce additional data from these systems as explained below.

TAO. As also explained in the March 7, 2022 submission, TAO is a distributed data store for the social graph. There is an object and associations to that object for all users. *See* Ex. A, Internal Facebook Wiki Regarding TAO Core Concepts. For each named plaintiff, Facebook will produce the user objects and associations to those objects.

Hive. As explained in prior submissions to the Special Master, Hive is Facebook's data warehouse. *See* Ex. B, Declaration of Mengee Ji In Support of Facebook Inc.'s Motion For Reconsideration of Special Master's Order Regarding Named Plaintiff Data, ¶ 11. [redacted] *Id.* ¶ 16. [redacted] *Id.* ¶ 17. [redacted] *Id.* ¶ 18.

[redacted] *Id.* These searches are subject to many technical and policy limitations which make searching for individual user data unduly burdensome, including that (1) [redacted] (*id.* ¶ 20); (2) [redacted] (*id.* ¶ 21); (3) [redacted] (*id.* ¶ 22); and (4) data in many Hive tables are in cold storage and would have to be restored in order to be searched and analyzed, *see* Ex. C, Declaration of Mengee Ji In Support of Facebook Inc.'s Motion For a Protective Order Against Production of API Call Logs, ¶¶ 17-18; Special Master's Nov. 8, 2021 Order Re: Facebook's Motion For Protective Order Against Production of API Call Logs, ¶ 15 ("Special Master Garrie finds that the data in the Mobile Table and Web Table is not reasonably accessible because it is not readily usable in its 'cold storage' state and must be restored to 'warm storage' in order to be searched and analyzed (i.e. usable).").

For these reasons, among others, as reported in its April 11, 2022 submission, Facebook estimates that it would take approximately [redacted] or around [redacted] of computational cost to search across the entire Hive data warehouse and extract all data about a single user.

Once the searches have been conducted, and data is returned, the data must be manually reviewed to, among other things, confirm that the data is associated with a particular user and does not include personal data of other users (e.g., user blocking another user) or trade secrets, and does not create system integrity or security concerns.

In light of the above technical limitations and burdens, Facebook proposes to search for and produce the following named plaintiff data from Hive to the extent it exists: (1) specific types of data requested or referenced by Plaintiffs in challenging Facebook's production of named plaintiff data; (2) a sample of data from Hive tables with user identifiers included in Exhibit B to the April 11, 2022 submission; and (3) the only data identified in Exhibit C to the April 11, 2022 submission that has not already been produced to Plaintiffs in the DYI file or otherwise: set permissions (audience controls on a post).

1. Specific Types of Data

- *Any remaining off-platform activity*: DYI includes off-platform activity. Facebook will search for and produce any underlying raw log-level data for off-platform activity provided to Facebook by a third party associated with the named plaintiffs.
- *Any remaining ad interests*: DYI includes ad interests. Facebook will search for and produce any underlying raw log-level data associated with the named plaintiffs.
- *Any remaining ad click data*: DYI includes ad click data. Facebook will search for and produce any underlying raw log-level ad clicks data associated with the named plaintiffs.
- *Ad impressions data*: Facebook will search for and produce ad impressions data associated with the named plaintiffs.
- *Any remaining custom audience data*: DYI identifies third parties who have created custom audiences associated with a user. Facebook will search for and produce any more granular information about custom audiences associated with the named plaintiffs, including custom audience type, whether it was used to deliver ads, and when.

2. Sampling from Hive tables identified in Exhibit A of April 11, 2022 submission

Given that data in Hive tables are not shared or made accessible to third parties, the significant burden of searching for and producing individual user data from Hive as explained in prior submissions, and that Facebook has already produced nearly a million pages of the named plaintiff data from the DYI system, Hive data is not relevant or proportional to the needs of this case. Facebook nevertheless understands that, as Judge Corley found in Discovery Order Nos. 11 and 12, Plaintiffs are allowed to test Facebook's positions on "sharing and accessibility" by obtaining discovery into what data exists and how it is used. To that end, and consistent with the requirements of relevance and proportionality under Rule 26, Facebook proposes a sampling process.

In addition to the above specific categories of data, Facebook will produce a sample of data in warm storage from Hive tables identified in Exhibit A of the April 11, 2022 submission. Specifically, Facebook will produce data from 200 tables in Exhibit A to be jointly selected by Facebook and Plaintiffs. Facebook will randomly select 100 tables and Plaintiffs will select 100 tables.

3. User data identified in Exhibit B of the April 11, 2022 submission and not in DYI

As shown in Exhibit C of Facebook's April 11, 2022 submission to the Special Master, only two types of user data in the contracts submitted to the Special Master for in camera review are not included in the DYI file: (1) privacy settings, and (2) set permissions (audience controls on a post). (1) has already been produced. Facebook will agree to produce (2) from TAO.

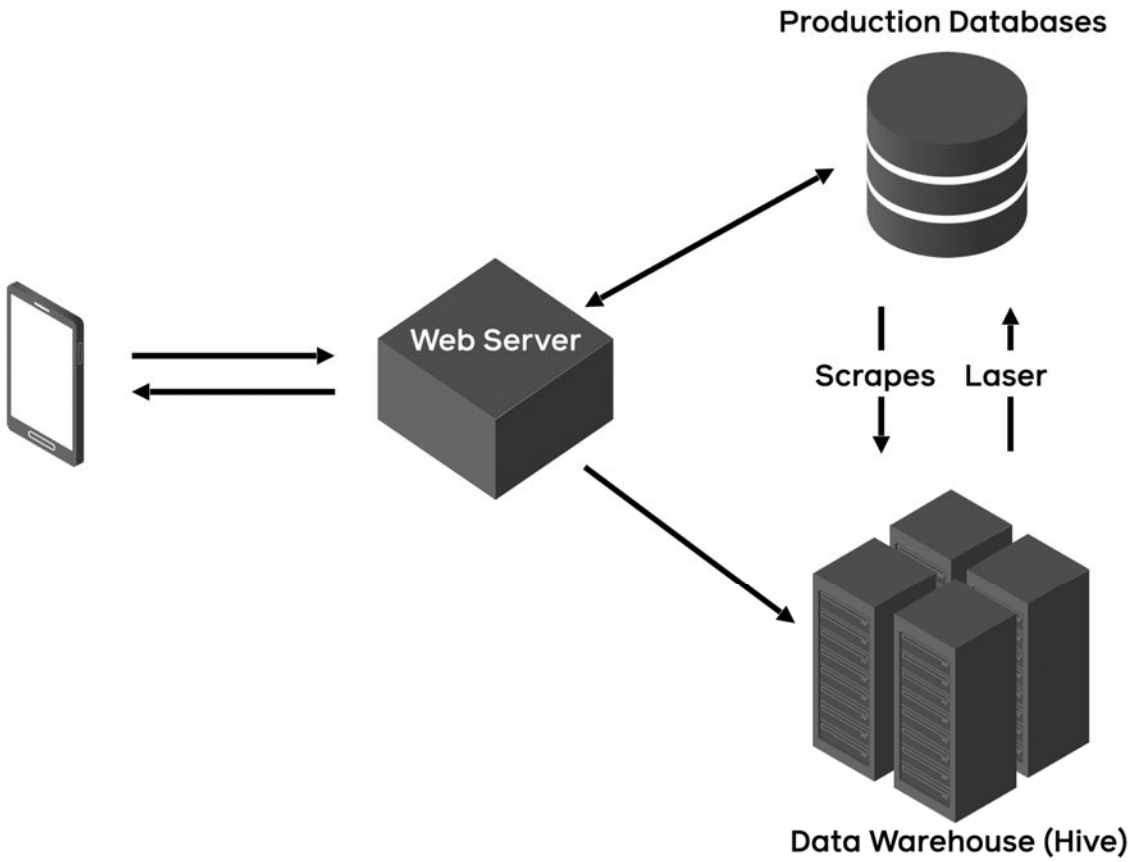
With respect to other Facebook systems, not including TAO and Hive, there are seven systems[1] that did not fall into one of the categories identified by the Special Master in Hearing Order Regarding Plaintiffs' Motion To Compel Production of Plaintiff Data, dated February 21, 2022: Ads Raw Storage, Manifold, orderdb (MySQL), Payments (MySQL), Tally, UDB, XDB, and ZippyDB. Producing data from any of these systems would be extremely burdensome and is not relevant or proportional to the needs of this case.

- *Ads Raw Storage and Manifold*: These systems are BLOB and binary storage and are not searchable by user identifiers.
- *Tally*: This system is a real-time aggregation counter and is not searchable by user identifiers.
- *orderdb (MySQL), Payments (MySQL), XDB, and Zippy DB*. These systems contain unstructured data and cannot be searched by user identifiers, except for tables using Ent or Node schemas for which there is a structured deletion plan. Searching other tables for named plaintiff data would require a full text search of all tables that would take many months to complete and generate a significant number of false positives requiring manual review of all data returned.

With respect to Dataswarm, the Special Master suggested that Facebook "query Tasks to identify Task Definitions which involve named plaintiff data" using identifiers,

[1] Callisto should have been put in Category (6) because the data is in DYI. UDB should also have been put in Category (6) because it is duplicative of TAO. MySQL describes the general use of MySQL databases and should not be a separate entry. For example, orderdb (MySQL) and Payments (MySQL) are specific uses of MySQL.

review those Tasks Definitions, and then search for named plaintiff data in the sources/destinations identified. We have investigated the Special Master's suggestion and did not include it in our proposed protocol because it is not feasible. Dataswarm is a collection of data processing operations coded in Python. Task Definitions are the Python code and Tasks are the running of that code. Searching Python code for user identifiers would be done through a syntax-only search which cannot understand the semantic meaning of the data being processed, i.e., whether a column name being processed involves a user identifier. As a result, additional code would have to be developed to understand whether/how identifiers are involved in a particular Task Definition.



Scenarios

No later than April 4, 2022, Facebook is to submit documentation sufficient to describe the data collected both on and off platform or provided by Third Parties in the following scenarios and provide written responses to the questions below: "Exhibit A to Plaintiff's Questions re: Data Collection and Use indicates that Facebook used predictive algorithms to generate five political segments for Facebook users (Very Liberal, Liberal, Moderate, Conservative, and Very Conservative) based on demographic, psychographic, and behavioral signals from Facebook user data."

1. What are the inputs into these algorithms (i.e. what are the demographic, psychographic, and behavioral signals used to generate the political segments)? Are these inputs provided by users or derived by Facebook?

Answer: As an initial matter, Very Liberal, Liberal, Moderate, Conservative, and Very Conservative are ad interests ("political ad interests"), which are distinct from political segmentation "based on demographic, psychographic, and behavioral signals from Facebook user data" ("political segments"). Political segments are political ad interests that are divided into political segments.

Political ad targeting segments were deprecated for new campaigns in January 2022 and for existing campaigns in March 2022. Based on a reasonable investigation to date, these were ad targeting segments derived from the following on-platform activity (user-provided data and observed data): [redacted]

Political segments were deprecated for new campaigns in October 2017 and for existing campaigns in January 2018. Based on a reasonable investigation to date, these segments were created by [redacted]

2. How are the psychographic signals computed (e.g. how is the psychographic signal "High Dollar Religious Donor" determined)?

Answer: Based on a reasonable investigation to date, we believe "High Dollar Religious Donor" was an ad targeting option based on partner categories. We do not know how this partner category was created. Partner categories were deprecated in October 2018.

3. Is information regarding identifiable ethnic affinities provided by users or derived by Facebook? How is ethnic affinity derived?

Answer: Ethnic affinities, also referred to as multicultural affinities, were ad interests created by Facebook based on a user's on-platform activity indicating an interest in content relating to certain communities. Ethnic affinity ad interests were deprecated in August 2020.

4. Where is political segmentation data for Facebook users stored?

Answer: Political segments were deprecated in October 2017 for new campaigns. Existing campaigns using political segments ended on January 1, 2018. Based on a reasonable investigation, we believe that political segments were deleted in January 2018.

5. Is political segmentation determined for a Facebook user as part of a data process that runs on a regularly scheduled basis or evaluated in real time when an ad is served?

Answer: Based on a reasonable investigation to date, we believe political segments were created manually by [redacted]

6. Is political segmentation associated with a Facebook user if possible (i.e. via UID, RID, SID, ASID, or another identifier that can be mapped to a user)? If so, explain how the political segmentation is associated with a Facebook user.

Answer: Based on a reasonable investigation to date, we believe political segments were associated with users by UIDs.

7. Is an individual's assigned political segment part of the DYI file?

Answer: Based on a reasonable investigation to date, we do not believe political segments were included in the DYI file.

Sincerely,

Rosemarie Ring

Rosemarie T. Ring

EXHIBIT A

Core Concepts

NOTE: This page has been deprecated; it has been moved to a new location.

TAO 101

This overview is primarily intended for new TAO engineers, PE-Cache engineers, and bootcampers, but it will be useful to anyone interested in TAO server design. The bulk of the document is a high-level look at the core components and architecture of assoc and FBObject caching in TAO. The miscellaneous topics page looks at failover handling, TAO configuration, deployment, priority gets, and more.

OTHER USEFUL RESOURCES

- TAO Ops wiki TAO configuration, administration, and troubleshooting
- TAO: The power of the graph Blog post with a useful history of TAO and caching at Facebook
- Infra Onboarding Slides
- TAO: Facebook's Distributed Data Store for the Social Graph - Conference paper.
- TAO Index page Most everything on wiki and Dex

Core Concepts

TAO is a high-performance service for storing, caching, and querying the graph of FBObjects and associations. It was developed to replace the original PHP logic that directly queried MySQL databases. It is highly optimized for large volumes of simple read queries.

Site operations rely overwhelmingly on the cache layer of objects and associations TAO provides. A typical Facebook page may aggregate hundreds of items from the social graph. TAO processes 100 trillion queries and 180 billion writes per day against a data set of many petabytes. During peak hours, TAO processes more than 1.4 billion reads a second and 2.5 million writes a second.

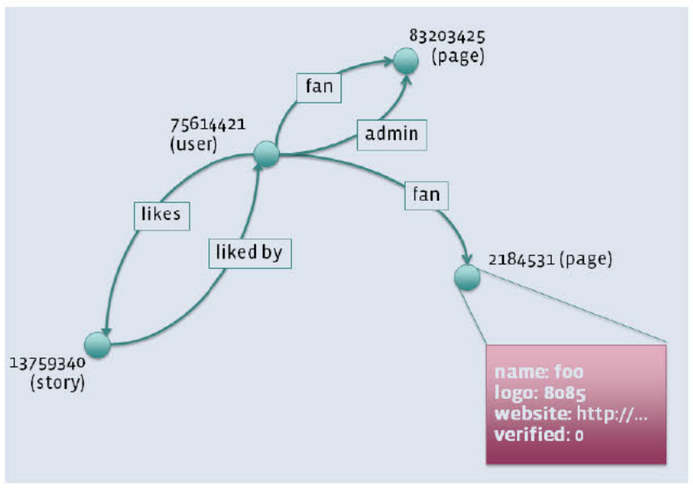
WRITE-THROUGH CACHE

TAO is a write-through cache (i.e., all writes to cache are simultaneously written to the backing store). This avoids the need to update cache through invalidation, yielding the following advantages:

- Improves cache hit rates on fast-growing assocs over our previous PHP-based implementation of assocs.
- Protect the UDB from thundering herds - all you need is pthread locks to serialize access to DBs.
- Support low-latency assoc writes from other regions - TAO can serve as a proxy on writes between service centers.
- Keep clients simple.

GRAPH ARCHITECTURE

TAO directly implements the graph abstraction of objects and associations that constitutes the social graph. Here's a tiny section of the graph; the nodes are FBObjects and the edges are associations:



TYPICAL HARDWARE

A typical TAO server has 144 GB RAM, 2 Intel Xeon 8 core E5-2660 CPUs running at 2.2Ghz with Hyper-Threading, and 10 Gigabit Ethernet, though other hardware configurations exist depending on region. TAO boxes in PRN use older hardware, and many TAO leaders in that region are supplemented with flash memory (see TaoDipper).

TAO is Forked from Memcache

To understand TAO at a high level, it's helpful to understand how it evolved. Very briefly, the Facebook web tier initially retrieved objects by generating MySQL queries directly to backing UDBs, which did not scale well at all. Because of the extremely high ratio of reads to writes, deploying a distributed in-memory caching layer was an obvious way to reduce request overhead. The initial caching scheme implemented was an in-house modified deployment of memcached, an open-source memory allocation system for caching. Memcached was designed for single servers, so our engineers developed a distributed architecture for Memcache based on MCProxy, a routing service that was responsible for managing TCP connections between boxes among other responsibilities.

Years later, TAO was developed as a specialized caching layer for objects and assocs, and for a variety of reasons, rather than writing it from the ground up, it was forked from Memcache. TAO has several key differences from Memcache (e.g., TAO uses a write-through architecture where Memcache is a demand-filled look-aside cache), but several core components of the system remain very similar, such as the internal request protocol and the memory structure. Memcache and TAO also use a similar channel to replicate invalidations across geographical regions, which relies on writing dirty queries to the backing db store and letting the MySQL replication channel handle some of the work.

Graph Elements

FBObjects

Client Information: FBObject

FBObjects are key/value pairs. The key is an FBid - a 64-bit ID space and generic sharding mechanism. The object is dictionary comprised by an arbitrary graph node or object (person, event, status update, et cetera). FBObjects can be pure ids with no associated value, though null values are not stored in memory.

FBObject objects can also be divided into the type configuration and the profile data.

The **type configuration** is a list of keys that defines what can be stored in a particular object. Type configs are indexed by integer keys called **FBTypes**, which define what kind of data is represented. For example, FBObject Type 2048 defines the USER object - you can view the scheme in TAOSchema here.

The **profile data** is the object itself - the set of valid attributes, as defined by the type configuration. They can be integers, strings, or arrays.

On the back end, keys are integers instead of strings (saves a lot of space and network bandwidth). We don't store empty attributes - empty strings or 0-values for integers.

Clients interact with FBObjects with the TAO API. Within TAO, get and set requests are formulated in a memcache-based protocol (see TAO Protocol section below).

Associations

Client Information: Associations

Assocs are directed edges between two FBids in the graph. All associations in production are served through TAO. Association, assoc, and edge are equivalent terms.

Assocs are characterized by **id1**, **id2**, and **type**. The id1 is the originating FBID, which connects to id2. The assoc type defines the relationship between id1 and id2 (id1 LIKES id2, et cetera).

Note: When TAO v1.15 is released, the assoc definition will allow for the use of an arbitrary 64-bit integer instead of an FBID.

Many associations entail an *inverse assoc*. Some assoc inverse associations are symmetrical, like *is friends with*. If Joan is friends with John, then John is friends with Joan. Other assoc inverse relationships are asymmetrical, such as *likes*. *John likes Porsche* does not entail *Porsche likes John.* The appropriate inverse type is *is liked by*: John *likes* Porsche, and Porsche *is liked by* John.

It is important to be clear on the type of inverse association implied by a particular association type - it will save you many tears. For more on assoc best practices, see Assoc Dos and Don'ts on Dex.

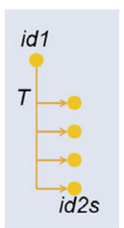
TAO allows association types to be configured with an inverse type when they're created, and provides support for keeping associations in sync with their inverses. Any creations, updates, and deletions are automatically coupled with the appropriate operation on the inverse association.

Assocs have two primary attributes: time (required) and data (optional). Despite the name, the *time* field is really an arbitrary sort key - it is populated with the time stamp by default. `Data` is an arbitrary field fixed at 16 bytes, beyond which it spills over to overflow buffer chains, illustrated here. Assoc data fields store a maximum of 255 bytes, although longer fields are supported in rare cases.

Like FBObjects, most assocs are defined by **type schemes** which specify the relationship represented by the edge (see Assocs with structured data). You can browse an extensive list of user-created assoc types in TAOSchema.

ASSOC LISTS

Assoc lists contain all edges of a given type originating from a given FBID. For example, an assoc list could list comments liked by a specific user, who would be the id1.



All assoc lists with the same id1 belong to the same TAO shard, are cached on the same TAO server; and are stored on the same UDB server.

ASSOC OPERATIONS

Several assoc operations are available in TAO:

- **add**, **remove**, or **change** an edge - TAO automatically applies appropriate operations to specified inverse assocs
- **range queries** - get a range of edges of a specific type originating from id1, e.g., "all friends of user A"
- **point queries** - get attributes of a specific edge between id1 and id2, if it exists
- **count queries** - number of edges of a given type originating from id1, e.g., "a given comment is liked by how many people?"
- **existence queries** - does an assoc of a specified id1, type, id2 exist?
- **cache invalidation** - remove from cache all edges of a specific type originating at id1

For information on how TAO caches and works with Assocs, see the Assoc Caching section below.

Tools

Caution: These tools may provide access to sensitive user information, and it is **imperative** when working with them that you follow Facebook's guidelines for user data access. Use good judgment.

A great way to understand fbobjects and assocs is to play around with the Entity Tool, which allows you to retrieve any entity of a specified ID.

You can browse FBObject and Assoc Types with the TAOSchema tool.

Cache Data Structure

Cache servers use a slab allocation scheme similar to memcache's, where dynamically allocated items are assigned to slab classes. Slab classes range from a minimum of 96 bytes to a maximum of MB. LRU eviction is performed within a slab class, but a background thread reallocates extents from one class to another to keep their eviction ages similar.

Objects and association lists are stored in a **single-chained hash table** illustrated here. Objects are indexed by the FBid, while association lists are indexed using an (id1,type) pair. We store a contiguous prefix of association lists in a time-ordered circular buffer, with an index by id2 if the list is long.

Each caching server further divides its slabs into **arenas** to prevent large, infrequently-requested objects and associations from evicting data of better types. Arenas are configured manually at start-up (details here).

Each arena maintains its own slab management scheme, but continues to link its items into the server's global hash table as before. In practice, our two most important arenas are for toxic types that are badly behaved and important types that should never be evicted. There are five arenas in total: fbobjects, toxic fbobjects, friends, friends replies, and a wildcard default arena.

We store small fixed-size items (e.g., association counts) separately, using direct-mapped 8-way associative caches that require no pointers. LRU order within each bucket is tracked by simply sliding the entries down.

TAODIPPER

Main article: TaoDipper; Dashboard

We're adding flash (FusionIO SSDs and LSI cards) to some TAO leaders to improve their performance as a redundant db cache. As of 2013, it's only being rolled out in Prineville, because that's where we have the tightest performance issues. The flash drives essentially form a new cache repository upstream from the RAM cache in the replication logic. RAM cache misses go to flash, and flash misses go to the UDB, with retrieved objects/assocs written to flash.

Related Services

mcrouter

Main Article: Mcrouter

mcrouter is a client-side daemon that handles routing and distribution of memcache/TAO/TACO requests for a host. It can also be run as the client library libmcrouter. mcrouter replaced mcproxy, which was fully removed from production in April 2013 - any references you may come across to mcproxy are old and should be updated.

TAOqueduct

TAO uses TAOqueduct to maintain cache consistency across regions. When TAO writes to a backing UDB in the master region, it adds a special 'memcache dirty' key to the MySQL binlogs of that UDB. For associations, the key is a TAO refill command, while for fbobjects, it is a TAO invalidate command. TAO then relies on cross-regional MySQL replication to transmit the key across regions, where it is read out of the binlogs in the form of invalidations in each replica region.

TAOqueduct is based on Aqueduct, which was previously used for invalidation distribution. It is substantially the same as Aqueduct, but has several differences, such as maintaining a far larger number of open connections between publisher and subscribers.

MCReplay2

If a TAO slave leader is unable to send an invalidation to followers, it logs them to local disk to be retried by mcreplay2.

TAO Umbrella

Main Article: TAO/Umbrella

TAO Umbrella is the current memcache communication protocol based on libmcrouter. It was developed to replace the previous libmcc-based implementation that required replies to be sent to clients in the order they were received, which caused head-of-line blocking problems. Umbrella allows for out-of-order replies, and in its initial deployment achieved a ~10% better throughput on cache-only get hits (under 100% CPU).

TACO

Main Article: TACO

TACO is essentially the same as TAO, except without the backing store. TACO mostly uses the same code paths as TAO. However, unlike TAO, TACO does send updates instead of invalidates and refills, and it broadcasts to all regions. Some teams find this useful because it provides proxy replication. TACO is good for write-heavy traffic in which occasional small losses of data are acceptable.

It is also suitable as an efficient list server (no disk IO). It supports very high write rates. **Data can be lost** so size appropriately.

Next>>

Category:TAO

« CLI Commands Full FBObject Caching »

tao

EXHIBIT B

GIBSON, DUNN & CRUTCHER LLP
Orin Snyder (*pro hac vice*)
osnyder@gibsondunn.com
200 Park Avenue
New York, NY 10166-0193
Telephone: 212.351.4000
Facsimile: 212.351.4035

Kristin A. Linsley (SBN 154148)
klinsley@gibsondunn.com
Martie Kutscher (SBN 302650)
mkutscherclark@gibsondunn.com
555 Mission Street, Suite 3000
San Francisco, CA 94105-0921
Telephone: 415.393.8200
Facsimile: 415.393.8306

GIBSON, DUNN & CRUTCHER LLP
Deborah Stein (SBN 224570)
dstein@gibsondunn.com
333 South Grand Avenue
Los Angeles, CA 90071-3197
Telephone: 213.229.7000
Facsimile: 213.229.7520

Joshua S. Lipshutz (SBN 242557)
jlipshutz@gibsondunn.com
1050 Connecticut Avenue, N.W.
Washington, DC 20036-5306
Telephone: 202.955.8500
Facsimile: 202.467.0539

Attorneys for Defendant Facebook, Inc.

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

IN RE: FACEBOOK, INC. CONSUMER PRIVACY USER PROFILE LITIGATION, This document relates to: ALL ACTIONS	CASE NO. 3:18-MD-02843-VC **DECLARATION OF MENGGE JI IN SUPPORT OF FACEBOOK, INC.'S MOTION FOR RECONSIDERATION OF THE SPECIAL MASTER'S ORDER RE PLAINTIFFS' MOTION TO COMPEL PRODUCTION OF PLAINTIFF DATA** Discovery Special Master Daniel Garrie, Esq.

DECLARATION OF MENGGE JI IN SUPPORT OF FACEBOOK, INC.'S MOTION FOR RECONSIDERATION OF THE SPECIAL MASTER'S ORDER RE PLAINTIFFS' MOTION TO COMPEL PRODUCTION OF PLAINTIFF DATA

I, Mengge Ji, declare as follows:

1. I am a Data Scientist at Meta, Inc. f/k/a Facebook, Inc. ("Facebook"). My job responsibilities include, among other things, understanding and working with Facebook's data systems, writing queries and conducting analyses of these data, researching Facebook's data and related technologies, and locating, analyzing, and exporting data for production in litigation and other legal matters. I submit this declaration in support of Facebook's motion for reconsideration of the Special Master's Order re Plaintiffs' Motion to Compel Production of Plaintiff Data. Unless otherwise stated, I have personal knowledge of the facts set forth herein, and, if called as a witness, I could and would competently testify thereto.

2. In my role as Data Scientist at Facebook, I am familiar with the general categories of user information Facebook maintains, where that data might be stored, and how they can potentially be accessed.

Data Contained in Facebook's Social Graph Related to Users' Activities on Facebook

3. Facebook users directly provide Facebook with data or information when using the Facebook platform. For instance, a user can upload a photo directly to her Facebook profile, which Facebook retains in order to display when that profile is accessed (if the user's privacy settings allow it). If the user "tags" a friend in that photo, Facebook also retains a record of which other Facebook user has been identified as appearing in that photo. In essence, the entire Facebook product that users interact with is a web of data points and relationships between data points that Facebook's systems present in a user-friendly format.

4. Facebook uses the term the "Social Graph" to describe this complex web of people, places, things, actions, and connections on the Facebook platform. As Facebook users navigate through Facebook and interact with it—including, for example, by commenting on

posts made by other users, watching videos, posting photos, and sending messages—the users create new relationships and connections between themselves and the content they are able to see. The activities a user takes on the Facebook platform, including posting pictures to their profiles, tagging friends in photos, and commenting on friends' timelines, are reflected in the Social Graph.

5. The Facebook product that users see is powered by a series of databases that work in tandem to provide Facebook users a seamless experience.[1] The key databases Facebook uses to support the Facebook product, which store the user content and information presently accessible via the Social Graph, are:

- [redacted]
- [redacted]
- [redacted]

[1] Facebook's Social Graph is not powered by a single database. Rather, Facebook is powered by an extraordinarily complex information architecture that stores information in various databases. The information in these databases is generally not human-readable and instead is intended to be processed for human consumption through Facebook's production environment.

- [redacted][2]

6. Between them, these databases contain trillions of data points provided to Facebook by its more than 2 billion users, many of which need to be readily accessible at all times so that users can view them as they use the Facebook product. As a result, the data infrastructure underlying the Social Graph is not only extraordinarily complex, but has also been specifically designed to serve the needs of the Facebook product.

7. One of the functions of the Facebook product is to be able to specifically identify user profiles or other sets of information associated with specific users. Put simply, when a Facebook user [redacted].

8. In order to support these and similar functions, the [redacted]

9. I understand that Facebook's "Download Your Information" or "DYI" tool retrieves data from and allows users to download a copy of data Facebook associates with their

[2] The Everstore database is in the process of being deprecated. The data contained in Everstore is being transitioned to Manifold.

Facebook account, including data associated with their account in the Social Graph. In order to compile this information, the DYI tool runs a set of searches that have been engineered by the DYI product team in order to pull data associated with that user from the Social Graph. The resulting data is presented in a user-readable format, rather than as code or strings of data.

Analytics Data Stored in Facebook's Data Warehouse ("Hive")

10. Facebook also stores data sets that it uses for internal analytics, product development, and other business functions. [redacted]

11. Facebook's internal analytics data is primarily stored in a data warehouse called Hive, which exists separately from the Social Graph. [redacted]

12. The Hive data warehouse is contained across a decentralized set of data centers, spread across the world. [redacted]

13. In addition to the Social Graph and Hive, I am aware of several other databases that contain data related to Facebook users, including data that is anonymized. I am not aware of any additional databases [redacted]

[redacted] As a result, searching these databases for individual user data would be a time-consuming, burdensome, manual, and iterative process akin to that described below.

14. I have also consulted with other members of the Data Science team regarding this question. My colleagues are also not aware of [redacted].

15. As detailed below, it would be extraordinarily burdensome to identify data associated with a specific user from data sources that are not [redacted].

Searching Within Hive

16. Unlike the Social Graph, Hive [redacted]. In other words, a Facebook data scientist [redacted].

17. [redacted] For instance, if a data scientist was endeavoring to [redacted] the data scientist would consult that

[redacted]. It is typically not efficient or productive to endeavor to just search randomly within Hive for categories or types of data.

18. Facebook does not have a process or tool that would allow it to [redacted]. Rather, doing so would be a [redacted]: a Facebook data scientist would need to [redacted]. This would be an extremely time-consuming process: I would first need to [redacted]. I would then need to [redacted] Depending on the size of the table, this search could take as little as a few minutes, or as much as 1-2 weeks, to run. Even if this search took only 1 hour to run per table, searching even [redacted] work by a Facebook data scientist, only to run the searches, not analyze or export any of the data.

19. By way of analogy, if Facebook were a library and each of the [redacted] were a book within that library, Facebook would not be able to search through all of the [redacted] Rather, Facebook would have to [redacted]. The search for data relating to a specific user within each of the [redacted] would be similarly manual and onerous. This would be a monumental undertaking.

20. Facebook is also limited in its ability to [redacted]

[redacted].

21. This search is also time-consuming due to the fact that many [redacted]

22. Many fields in Hive tables [redacted]. As a result, if the information needs to be used or analyzed [redacted] This is an extra step of analysis that would need to be conducted if the results of any searches for specific user IDs would need to be produced to the Plaintiffs. This would add between a few minutes and a few days of work to each search that yielded results.

23. Even after [redacted] the output of searching the Hive table would be primarily [redacted] and difficult to interpret. Within Facebook, [redacted]. Nor could the results of [redacted] be easily consolidated. Because the search is [redacted], the results would be output in at least [redacted]

Exporting Data from Hive

24. Even if Facebook were to spend several years searching for all data points associated with specific user IDs, [redacted].

25. For Facebook to [redacted]). Particularly large datasets would also need to be broken into small pieces prior to being decompressed and written [redacted]. This would be a process conducted by Facebook data scientists on the legal team. This manual, iterative process introduces the possibility of human error, which necessitates real-time quality checks of the data downloads that take additional time and create the potential for delay.

26. First, a Facebook data scientist must [redacted]

27. Second, [redacted] The speed at which Facebook can export the data from the warehouse to the server varies depending on usage of its systems in the ordinary course of business, but based on testing and prior exports, Facebook's current best estimate is that it takes about [redacted]

28. Third, [redacted], a data scientist must [redacted] The process must be repeated over and over until the export is complete.

29. Preparing any productions also requires substantial additional resources from senior members of the eDiscovery data scientist team to prepare for and manage the coding and export project. Moreover, after the initial exported files have been created, it would take additional time for a review team to validate the queries and data and perform additional quality control, or longer if there are issues that require remediation.

30. In total, Facebook currently estimates that, even without any other responsibilities or deadlines, it would be a substantial [redacted] through the process described above. The Facebook eDiscovery team, which is responsible for assisting Facebook in-house and outside counsel in active litigations and other legal matters, in addition to building and maintaining internal infrastructure crucial to the management and preservation of data on legal hold, does not have the time and resources required to search for, access, analyze, and export the data in millions of Hive tables in the manner described above.

31. Facebook cannot materially shorten this timeline by hiring new employees because it would take time and resources to interview, select, and onboard new employees, and any new hires would have to be sufficiently trained regarding Facebook's systems, policies, and procedures, which is a mandatory process that takes three weeks. For the same reason, Facebook cannot simply engage third party consultants or temporary employees to handle this data export. Nor would adding more servers—which would require diverting them from their use in the ordinary course of business—necessarily reduce the estimated timeline in a linear fashion, as the export still requires [redacted]. All of these options—hiring new employees, hiring contractors, and adding servers—would also be extremely costly.

I declare under penalty of perjury that the foregoing is true and correct, and that I executed this Declaration on December 10, 2021, in Sausalito, California.

Mengge Ji

EXHIBIT C

GIBSON, DUNN & CRUTCHER LLP
Orin Snyder (*pro hac vice*)
osnyder@gibsondunn.com
200 Park Avenue
New York, NY 10166-0193
Telephone: 212.351.4000
Facsimile: 212.351.4035

Kristin A. Linsley (SBN 154148)
klinsley@gibsondunn.com
Martie Kutscher (SBN 302650)
mkutscherclark@gibsondunn.com
555 Mission Street, Suite 3000
San Francisco, CA 94105-0921
Telephone: 415.393.8200
Facsimile: 415.393.8306

GIBSON, DUNN & CRUTCHER LLP
Deborah Stein (SBN 224570)
dstein@gibsondunn.com
333 South Grand Avenue
Los Angeles, CA 90071-3197
Telephone: 213.229.7000
Facsimile: 213.229.7520

Joshua S. Lipshutz (SBN 242557)
jlipshutz@gibsondunn.com
1050 Connecticut Avenue, N.W.
Washington, DC 20036-5306
Telephone: 202.955.8500
Facsimile: 202.467.0539

Attorneys for Defendant Facebook, Inc.

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

IN RE: FACEBOOK, INC. CONSUMER PRIVACY USER PROFILE LITIGATION, This document relates to: ALL ACTIONS	CASE NO. 3:18-MD-02843-VC **DECLARATION OF MENGGE JI IN SUPPORT OF FACEBOOK, INC.'S MOTION FOR A PROTECTIVE ORDER AGAINST PRODUCTION OF API CALL LOGS** Discovery Special Master Daniel Garrie, Esq.

DECLARATION OF MENGGE JI IN SUPPORT OF FACEBOOK, INC.'S MOTION FOR A PROTECTIVE ORDER AGAINST PRODUCTION OF API CALL LOGS

HIGHLY CONFIDENTIAL

I, Mengge Ji, declare as follows:

1. I am a Data Scientist at Facebook, Inc. ("Facebook"). My job responsibilities include, among other things, understanding and working with Facebook's data systems, writing queries and conducting analyses of these data, researching Facebook's data and related technologies, and locating, analyzing, and exporting data for production in litigation and other legal matters. I submit this declaration in support of Facebook's motion for a protective order as against the production of certain of Facebook's API call logs. Unless otherwise stated, I have personal knowledge of the facts set forth herein, and, if called as a witness, I could and would competently testify thereto.

2. My understanding is that Plaintiffs in this matter have requested that Facebook produce to them all "logs of API Calls made by Third Parties for the Content and Information of Friends of Installing Users," including specific information about, among other things, "the response status of those Calls (*e.g.*, success or reason for failure)" and "[t]he Content and Information delivered to Third Parties on each API in response to these Calls."

3. I also understand that Plaintiffs have issued additional requests for information relating to the number of API calls made by certain "Third Parties" and "Business Partner[s]," including "the number of calls . . . made to each such API . . . each month" and "the volume of data transferred from each such API . . . each month."

1. Background on Facebook's APIs and Call Logs

4. Application Program Interfaces ("APIs") are standard computing protocols used throughout the digital world. Each time a user visits an app, the user necessarily interacts through an API. And every company that allows third parties to communicate with their servers

uses APIs. As just one example, email programs like Outlook or Mail on iOS communicate with the user's email provider (Yahoo, Gmail, etc.) through APIs to obtain the user's messages.

5. Facebook has a series of APIs that allow for data to be transferred to and from Facebook's Platform. The Facebook Platform is powered by a series of databases that Facebook relies upon to host and run the Facebook products and which work in tandem to provide Facebook users a seamless experience as they access different categories and types of content posted on the Platform, by themselves or other users. APIs facilitate the transfer of data from the databases that support the Platform to websites and applications, including Facebook's own, proprietary mobile apps. For instance, if a Facebook user pulls up her profile photo via the Facebook mobile app, the Facebook mobile app will "call" a Facebook API to request that photo to be displayed locally on the user's phone.

6. Third party mobile apps and websites can also connect to Facebook through its APIs. For instance, if a user chooses to log in to NYTimes.com with his Facebook account instead of creating an account for that site, NYTimes.com will call the Facebook APIs to verify the user's identity. Facebook has a series of public APIs that are published on its developer website (https://developers.facebook.com/), and app developers can use them or request to use them to send or receive information from the Facebook Platform.

7. Facebook maintains internal logs that contain the number of API calls made to the Facebook Platform. Two of the data tables that contain [redacted] information Facebook maintains regarding API call activity are the [redacted] table and the [redacted] table.

2. The [redacted] and [redacted] Tables

8. The [redacted] table contains logs of API calls [redacted]

[redacted] The data in the [redacted] table dates back to [redacted] As of October 6, 2021, Facebook had [redacted] of data in this table.

9. The categories of data logged in the [redacted] table include, among other things, [redacted] A true and correct list of the data fields reflected in the [redacted] table is attached hereto as **Exhibit A**. A true and correct sample of [redacted] from the [redacted] table, with [redacted] redacted, is attached hereto as **Exhibit B**.

10. The [redacted] table contains logs of API calls [redacted] The data in the [redacted] table dates back to [redacted] As of October 6, 2021, Facebook had [redacted] of data in this table.

11. The categories of data logged in the a[redacted] table include, among other things, [redacted] A true and correct list of the data fields reflected in the [redacted] table is attached hereto as **Exhibit C**. A true and correct sample of ten lines of data from the [redacted] table, with the [redacted] redacted, is attached hereto as **Exhibit D**.

12. Several of the fields in each of these tables do not contain [redacted]

[redacted] This is an extra step of analysis that would need to be conducted if these tables were ever to be used or analyzed outside of Facebook's environment.

13. The [redacted] and [redacted] tables do not contain data demonstrating whether data was returned in response to an API call, what the data that was returned may have been (if it was returned at all), or the volume of data that may have been returned (if it was transferred at all).

14. I understand that there are many reasons why data would not be returned in response to an API call. These reasons include, but are not limited to, [redacted]

A. Storage of the [redacted] and [redacted] tables

15. The [redacted] and [redacted] tables are stored in a highly compressed state in Facebook's internal decentralized, structured data warehouse, known as "Hive." Hive is designed to ingest and process data for internal analysis, not to export data into files that can be transferred outside of Facebook. As discussed further below, accessing and analyzing large data tables in Hive is a multi-step process that can take a considerable amount of time, depending on the volume of data being accessed and the complexity of the analysis or querying needed.

16. The active [redacted] and [redacted] tables are currently maintained pursuant to [redacted] business retention periods [redacted]

[redacted]

17. [redacted]

18. Tables in cold storage cannot be reviewed or analyzed while they remain in cold storage. If Facebook wants to conduct any analysis of data in cold storage, a Facebook data scientist must first restore it from cold storage to an accessible location, referred to as "warm" storage. It takes approximately 1 day of processing to restore [redacted] of cold storage data. Restored cold storage data is only made available to legal team members.

19. The [redacted] and [redacted] tables are stored in "partitions" or segments. These tables are partitioned [redacted] Because each [redacted] is partitioned off from the others, Facebook cannot quickly or efficiently run searches across multiple [redacted] Instead, Facebook's system is optimized for running searches within an individual partition. As a result, Facebook cannot search these entire tables for a specific value, for instance a [redacted] while in cold storage, and instead must restore individual partitions to warm storage and then search the partitions individually. Due to the same limitations on accessing and searching multiple partitions of data, Facebook also cannot filter these tables for [redacted] without first restoring partitions from cold storage piecemeal.

20. Accordingly, to search or analyze the data in either the [redacted] table or [redacted] tables, a Facebook data scientist must first restore a partition of the data to warm

storage. Due to limitations on computing and server capacity, only a limited number of partitions can be restored and analyzed at any given time.

21. The [redacted] and [redacted] tables are very large data tables, even in comparison to other data sets at Facebook and even when compressed in Facebook's specialized data storage environment. The volume of the [redacted] table is currently [redacted] in its most highly-compressed form in cold storage. The volume of the [redacted] table is currently [redacted] in its most highly-compressed form. Maintaining the high volume of this data [redacted] is beginning to tax Facebook's data storage system, which is one of the largest and most sophisticated data storage systems in the world. For instance, I understand that it is costing Facebook approximately [redacted] to store the [redacted] data in its compressed form.

B. Sampling of the [redacted] table

22. In order to assess how long it might take to produce data from the [redacted] table to the Plaintiffs in this action, I recently restored and analyzed one partition of data from the [redacted] dated [redacted]

23. The volume of the [redacted] partition is [redacted] in cold storage. For reference, 1 TB is four-times the size of a standard laptop computer hard drive of 256 GB, so the *compressed* volume of the data for this one day equals [redacted] The size of the data in each partition in the [redacted] table varies day [redacted] For example, the most recent partitions of the table in cold storage are around [redacted] each.

24. In order to analyze the [redacted] data, I first restored this partition to warm storage. The length of time it takes to restore data from cold storage depends on how many

requests are pending at the same time and the computing power available, which varies depending on the time of day. In this instance, it took approximately 2 hours of processing time to restore this single partition into warm storage. Based on the volume of this partition and others in this table, I estimate that Facebook presently has the computing and server capacity to host up to five partitions of this table in warm storage at a time. As a result, Facebook can analyze no more than five partitions of this table at a given time in the ordinary course. In other words, an effort to simultaneously analyze more than five partitions of this data would displace and disrupt all other efforts by the legal team to perform other data analyses for this and other matters, including for the [redacted] table.

25. The [redacted] partition of the [redacted] table contains [redacted] As a point of reference, a Microsoft Excel file can only hold up to 1 million rows of data. If translated into Excel, then, this one partition of the [redacted] table would comprise more than [redacted] Excel files.

26. As I mentioned in paragraph 8 above, this table only contains logging for API calls [redacted] The partition only contained data for [redacted] It took 1 hour and 14 minutes for this search to complete.

27. [redacted]

28. Based on the results of this exercise, I estimate it would take 2,378 hours [redacted] in cold storage) to restore each of the partitions in the [redacted] table to warm storage and additional 1,279 hours [redacted] to run a single search on each. Assuming I worked exclusively on this task for 8-hours per day, 260 days each year, it would take nearly two years for me to run this same search on [redacted] of this data (through the [redacted] partition). This is also a conservative estimate, [redacted] This estimate also does not account for the other demands on Facebook's personnel and resources (including the need to use these same systems to respond to requests for other legal matters), occasional outages, or many other variables.

C. Sampling of the [redacted] table

29. I also restored and recently analyzed one partition of data from the [redacted] table dated [redacted]

30. The volume of the [redacted] partition is [redacted] in cold storage. The size of the data in each partition in the [redacted] table varies [redacted]. For example, the most recent partitions of the table in cold storage are around 40 TB each.

31. In order to analyze the [redacted] data, I first restored this partition to warm storage. The length of time it takes to restore data from cold storage depends on how many requests are pending at the same time and the computing power available, which varies depending on the time of day. In this instance, it took approximately [redacted] hours of processing time to restore this partition into warm storage for analysis. Based on the volume of this partition and others in this table, I estimate that Facebook presently has the computing and server

capacity to host 20 partitions of this table in warm storage at a time. As a result, Facebook can only analyze 20 partitions of this table at a given time in the ordinary course.

32. The [redacted] partition of the [redacted] table contains [redacted] rows of information. If exported to Excel, then, this partition of the [redacted] table would alone comprise [redacted] Excel files.

33. I ran a search for [redacted] The partition only contained data for [redacted] It took 43 minutes for this search to run.

34. Of the [redacted] of data in this table, [redacted] of data were returned for these [redacted]

35. Based on the results of this exercise, I estimate it would take 2,970 hours (2.5 hours per partition in cold storage) to restore each of the partitions in the [redacted] table to warm storage and an additional 935 hours (45 minutes per partition) to run a single search on each. Assuming I worked exclusively on this task for 8-hours per day, 260 days each year, it would take nearly two years for me to run this same search on each partition of this data (through the [redacted] partition). This is also a conservative estimate, [redacted] This estimate also does not account for the other demands on Facebook's personnel and resources, occasional outages, or many other variables. This estimate is also *in addition to* the two years it would take to search the [redacted] table.

36. As a result, it is neither feasible nor practicable to restore all of the partitions of the [redacted] and/or [redacted] tables at the same time and [redacted]

3. The Burden of Production of API Call Log Data

37. As noted, the [redacted] and [redacted] tables are stored in Facebook's internal structured data warehouse, known as "Hive." This data warehouse is not optimized to export data in a manner that can be transferred outside of Facebook. For Facebook to export and produce data from Hive tables, the data must be decompressed, broken into small pieces, and re-written to .csv files (which can be opened with Excel). This would be a manual, iterative process conducted by Facebook data scientists on the legal team. As a result, the timeline for completing an export needs to account not only for the limitations on Facebook's computing capacity, the size limitations on our mechanisms for export, and the availability of Facebook's technical resources, but also the possibility of human error, which necessitates real-time quality checks of the data downloads.

38. Before any analysis or export can be conducted, the data must be transitioned to warm storage in order to be reviewed, analyzed, or exported. As discussed above in Paragraphs 28 and 35, I estimate it would take 5,348 hours to restore all of the partitions of [redacted] and [redacted] tables in to warm storage, which is more than 2.5 years of work, assuming I work exclusively on this task for 8 hours per day, 260 days each year, and do not run any searches to narrow the data set. Running searches on the data set would take at least an additional year.

39. After data has been restored into warm storage, it is necessary to decompress the data, which requires that Facebook take several additional steps, which are themselves time-consuming.

40. First, a Facebook data scientist must write entirely new code to query the data and demarcate each partition into chunks of data that can be effectively transported to and/or received by the recipient. For instance, if the recipient intends to use Excel to analyze the data, each partition must be broken into chunks of 1 million rows each, which is the maximum number of rows that can be opened in an Excel file. When Facebook must produce Hive data in litigation, it typically produces it in Excel form. If the intended recipient—in this instance, Plaintiffs—had access to or intended to use specialized software to access the data (*e.g.*, SQL, R, Stata, SAS) and had sufficient computing power to access larger files, the limiting factor would be a combination of size of the drive used to transport the data and the maximum number of rows that could be reliably written into a .csv and reviewed for quality control. I understand from my colleagues at Facebook that the largest .csv files that we have handled and transferred reliably is 50 GB, which is roughly [redacted]

41. Second, the code is run to export that data to a server, which would involve querying each chunk of [redacted] from the data warehouse and then writing the results of that query as .csv files in a decompressed flat-file format. The speed at which Facebook can export the data from the warehouse to the server varies depending on usage of its systems in the ordinary course of business, but based on testing and prior exports, Facebook's current best estimate is that it takes about 40 minutes to write a 50 GB .csv file. At this rate, it would take approximately 15 hours to extract the [redacted] of data I identified in my

search of the [redacted] partition of the [redacted] table [redacted] discussed in Paragraph 27 above, to .csv format. The resulting .csv files would be approximately [redacted] total, or approximately f[redacted]. It would then take our eDiscovery and Data Science teams about 3 minutes per file to inspect the data to confirm that it was not corrupted or truncated during export. If we were to export the [redacted] of data I identified in my search of the [redacted] partition of the [redacted] table—[redacted], it would take approximately 1.5 hours to inspect these files. It would therefore take approximately 16.5 hours to extract and inspect .csv files containing [redacted]

42. It would separately take 2 hours to extract the [redacted] data I identified in my search of [redacted] discussed in Paragraph 34 above, to .csv format. The resulting .csv files would be approximately [redacted] total. It would then take our e-Discovery and Data Science teams about 3 minutes per file to inspect the data to confirm that it was not corrupted or truncated during export. If we were to export the [redacted] data I identified in my search of the [redacted] partition of the [redacted] it would take approximately 12 minutes to inspect these files. It would therefore take an additional 2 to 3 hours to extract and inspect .csv files containing [redacted]

43. Third, once the server is full, a data scientist must empty the server by transferring the .csv files into a local encrypted hard drive or a secure transfer site. Each 50 GB .csv file

would require 5 to 8 hours to be transferred from the Facebook servers to a physical hard drive, depending on network speeds and other variables.

44. Decompression of the data, which is required for export of this data, results in a significant expansion in the size of the data. Within Facebook's systems, one partition of the [redacted] table comprises approximately [redacted] of data and one partition of the **api_hits_www** table comprises approximately [redacted] of data. It is difficult to estimate the compression ratio before the data is actually exported. Based on testing and prior exports, Facebook's current best estimate as to how big a single partition of each of these tables would be when fully decompressed to .csv format for export is around [redacted] per partition for the [redacted] table and [redacted] per partition for the [redacted] table.

45. [redacted] As a result, the second and third steps of this process would have to be run a minimum of 19,250 separate times for each partition of the [redacted] table. Similarly, the second and third steps of this process would have to be run a minimum of 4,750 separate times for each partition of the [redacted] table. It is not possible to expand the number of available servers without investing significantly in additional resources or interrupting other business and legal functions.

46. The decompressed volume of a single partition of data from each of these tables—up to [redacted] table and [redacted] for the [redacted] table—is significantly larger than any hard drives or secure transfer site that Facebook has access to. Facebook's current best estimate is that exporting a single partition of the [redacted] table would require it to load .csv files onto [redacted] encrypted hard drives that would cost approximately $1,200 each, presuming Facebook can acquire a

sufficient number of them. Facebook's current best estimate is that exporting a single partition of the [redacted] table would require it to load .csv files onto [redacted] encrypted hard drives that would cost approximately $1,200 each, again presuming Facebook can acquire a sufficient number of them.

47. The process must be repeated over and over until the export is complete. Facebook estimates that it would take 8,000 days and $70,000 in hardware costs to export the smallest, single partition of the [redacted] table and 2,000 days and $17,000 in hardware costs to export a single partition of the [redacted] table.[1] Replicating this process for more than [redacted] of each table would multiply those timing estimates proportionally.

48. This manual, iterative process introduces the possibility of human error, which necessitates real-time quality checks of the data downloads that take additional time and create the potential for delay.

49. Facebook also estimates that it will take several days for senior members of the eDiscovery data scientist team conducting the export to prepare for and manage the coding and export project. After the initial exported files have been created, Facebook estimates it would take an additional month for a review team to validate the queries and data and perform additional quality control, or longer if there are issues that require remediation.

A. Impact on Facebook's E-Discovery Team's Resources

50. In total, Facebook currently estimates that, even without any other responsibilities or deadlines, it would be substantial, multi-year project to attempt to export the [redacted] table and the [redacted] table through the process described above. The Facebook E-Discovery team, which is responsible for assisting Facebook in-house and outside counsel in

[1] The shipping costs for the hard drives would add additional costs onto this project.

active litigations and other legal matters, in addition to building and maintaining internal infrastructure crucial to the management and preservation of data on legal hold, does not have the time and resources required to access, analyze, and export the data in the [redacted] table and the [redacted] table in the manner described above.

51. Facebook cannot materially shorten this timeline by hiring new employees because it would take time and resources to interview, select, and onboard new employees, and any new hires would have to be sufficiently trained regarding Facebook's systems, policies, and procedures, which is a mandatory process that takes three weeks. The API call log data export in particular requires specialized knowledge about Facebook's systems because the system is not built for the purpose of exporting the extremely large amounts of data at issue—knowledge that non-Facebook personnel do not have, given the nature of the proprietary systems and databases employed by Facebook. For the same reason, Facebook cannot simply engage third party consultants or temporary employees to handle this data export. Nor would adding more servers—which would require diverting them from their use in the ordinary course of business—necessarily reduce the estimated timeline in a linear fashion, as the export still requires downloading the data from the server, and the manual process of writing the code and monitoring the export. All of these options—hiring new employees, hiring contractors, and adding servers—would also be extremely costly.

4. Alternative Production Proposal

52. I understand that Facebook has proposed to produce summary data relating to API calls to Plaintiffs in this matter instead of data from the [redacted] and [redacted] tables. [redacted]

[redacted]

53. Specifically, I understand that Facebook has proposed to produce data from the [redacted] table. The [redacted] table contains summaries of data [redacted]

54. Because the [redacted] table summarizes [redacted] it is substantially less burdensome to store and analyze.

55. The [redacted] table contains summary data regarding API calls made to the Facebook Platform, [redacted] The fields contained in this table include, among other things, [redacted] A true and correct list of the data fields reflected in the [redacted] table is attached hereto as **Exhibit E**. A true and correct sample of [redacted] from the [redacted] table is attached hereto as **Exhibit F**.

56. The data in the [redacted] table dates back to [redacted]

57. The [redacted] table is maintained in warm storage in the Hive data warehouse. It is currently [redacted] in this compressed form. A single partition [redacted]

of this table is around [redacted] in compressed form and [redacted] in raw (*i.e.*, not compressed) form.

58. It is difficult to estimate the compression ratio before the data is actually exported. Based on testing and prior exports, Facebook's current best estimate as to how big an export of data from this table dating from [redacted] would be when fully decompressed to .csv format for export is [redacted]

59. Facebook is currently preparing an export of data from this table dating from [redacted] for production to Plaintiffs in this matter and estimates it will be able to produce this data by November 30, 2021.

60. A brief summary of the three tables discussed herein is provided below, along with relevant data points regarding the burden of production.

	[redacted]	[redacted]	[redacted]
Earliest Date	[redacted]	[redacted]	[redacted]
Compressed Size	[redacted]	[redacted]	[redacted]
Est. Decompressed Partition Size	[redacted]	[redacted]	[redacted]
Est. Decompressed Size of Data [redacted] 12/31/19	[redacted]	[redacted]	[redacted]

I declare under penalty of perjury that the foregoing is true and correct, and that I executed this Declaration on October 18, 2021, in Sausalito, California.

Mengge Ji

Exhibit A

to Ji Declaration

Highly Confidential

api_hits_mobile

Column	Type	Description
ds	string	Date on which the data were generated
fb_mobile_app	string	The mobile app that the user was using
ts	string	Time stamp showing hour on which the data were generated
session_origin	bigint	The install flow from which the session originated
userid	bigint	The result of calling getUserID on a supplied ViewerContext. This could be a person's ID, a page's, oran application's. If you only want the person behind the page or app, you want the 'accountid' field.
carrier	string	The name of the carrier that the user's phone is using.
impersonating_appid	bigint	App_id that was being impersonated
ip_address	string	The IP address from which the user actually accessed Facebook.
method_type	bigint	Category of the API method
user_agent	string	The UserAgent field from a user's browser
accountstatus	bigint	The result of calling getConfirmedValue on the EntAccount inside the ViewerContext
appid	bigint	Application ID
event_time	double	The webserver timestamp at the moment that log() was called.
bandwidth_usage	bigint	Number of bytes in the response
is_connect_site	boolean	Whether the connection came from Facebook Connect
duration	bigint	Response time for the API call
device	struct<model:string>	Structure of data representing info about a mobile device
proxy_ip_address	string	RemoteAccess::isProxied, if exists
mobile_hashid	string	Hive alias for device_instance_id
method	string	The API method that was logged.
memory_usage	bigint	Maximum amount of memory used during the request
error_code	bigint	Error code thrown
bg_ping	boolean	Was the request made while app was backgrounded?
rid_for_userid	bigint	The Clementine RID for this UserID, if exists.
sdk_type	bigint	Which Facebook SDK was used to make this call
environment	bigint	The TFBEnv value.
mobile_client_locale	string	The locale being used on the mobile client, determined from useragent
predicted_network_provider_id	bigint	From deserializer (ORPHANED at 2014-12-12 15:33:13)
predicted_connection_type	string	Our prediction of whether the connection is via wifi, or cellular.
is_shadow_req	boolean	From deserializer (ORPHANED at 2014-12-12 15:33:13)
predicted_carrier_label	string	Our prediction of the carrier, or 'network provider', that the mobile device is operating on.
brand	string	Our prediction of the brand of mobile device
client_country_code	string	The country code of the mobile client
predicted_carrier_id	bigint	From deserializer (ORPHANED at 2014-12-12 15:33:13)
target_id	string	Graph API ID of the root node of the request
ip_carrier_id	bigint	Our legacy ID that predicts the carrier that the mobile device is operating on, based on IP address.Matches the 'carrier' field from api_hits
applied_api_version	string	API version that was effectively applied to this request
query_name	string	The friendly name, if one exists. Otherwise if this is an FQL request, the FQL query hash.
preview_mode	boolean	Was the request made while app is in preview mode?
dialtone_mode	boolean	Was the request made while app is in dialtone mode?
enabled_features	array<string>	A set of features that have been enabled for this hit.
fbtrace_id	string	Fbtrace ID, unique for each request
communityid	bigint	Workplace Community ID
accountid	bigint	The logged in user ACCOUNT's ID. (Different from UserID, which could be a page.)
ip_address_unsafe	string	The IP address from which the user claims to have accessed Facebook.
rid_for_accountid	bigint	The Clementine RID for this UserID, if exists.
is_developer_site_call	boolean	Did the request originate from developers.facebook.com
www_request_id	string	Request ID, unique for each request

Exhibit B

to Ji Declaration

Highly Confidential

api_hits_mobile Sample

session_origin	userid	carrier	impersonating_appid	ip_address	method_type
0	Redaction - UII	1.83E+14	0	Redaction - UII	0
0		1.83E+14	0		0
0		1.83E+14	0		0
0		6.32E+14	0		0
0		6.32E+14	0		0
0		7.31E+14	0		0
0		7.31E+14	0		0
0		7.31E+14	0		0
0		6.32E+14	0		0
0		1.83E+14	0		0

user_agent
SupportsFresco=1 Dalvik/2.1.0 (Linux; U; Android 10; SM-J600G Build/QP1A.190711.020) [FBAN/EMA;FBBV/129523958;FBAV/115.0.0.9.100$FB4A$;FBDV/SM-J600G;FBLC/en_US;FBNG/4G;FBMNT/METI
SupportsFresco=1 Dalvik/2.1.0 (Linux; U; Android 10; SM-J600G Build/QP1A.190711.020) [FBAN/EMA;FBBV/129523958;FBAV/115.0.0.9.100$FB4A$;FBDV/SM-J600G;FBLC/en_US;FBNG/4G;FBMNT/METI
SupportsFresco=1 Dalvik/2.1.0 (Linux; U; Android 10; SM-J600G Build/QP1A.190711.020) [FBAN/EMA;FBBV/129523958;FBAV/115.0.0.9.100$FB4A$;FBDV/SM-J600G;FBLC/en_US;FBNG/4G;FBMNT/METI
SupportsFresco=1 Dalvik/2.1.0 (Linux; U; Android 7.0; Y108 Build/NRD90M) [FBAN/EMA;FBBV/118379517;FBAV/102.0.0.12.163$FB4A$;FBDV/Y108;FBLC/en_US;FBNG/WIFI;FBMNT/METERED;FBDM/{de
SupportsFresco=1 Dalvik/2.1.0 (Linux; U; Android 7.0; Y108 Build/NRD90M) [FBAN/EMA;FBBV/118379517;FBAV/102.0.0.12.163$FB4A$;FBDV/Y108;FBLC/en_US;FBNG/WIFI;FBMNT/METERED;FBDM/{de
SupportsFresco=1 Dalvik/2.1.0 (Linux; U; Android 7.1.1; Moto E (4) Build/NMA26.42-169) [FBAN/EMA;FBBV/99791899;FBAV/87.0.0.0.90$FB4A$;FBDV/Moto E (4);FBLC/en_US;FBNG/WIFI;FBMNT/NOT_I
SupportsFresco=1 Dalvik/2.1.0 (Linux; U; Android 7.1.1; Moto E (4) Build/NMA26.42-169) [FBAN/EMA;FBBV/99791899;FBAV/87.0.0.0.90$FB4A$;FBDV/Moto E (4);FBLC/en_US;FBNG/WIFI;FBMNT/NOT_I
SupportsFresco=1 Dalvik/2.1.0 (Linux; U; Android 7.1.1; Moto E (4) Build/NMA26.42-169) [FBAN/EMA;FBBV/99791899;FBAV/87.0.0.0.90$FB4A$;FBDV/Moto E (4);FBLC/en_US;FBNG/WIFI;FBMNT/NOT_I
SupportsFresco=1 Dalvik/2.1.0 (Linux; U; Android 7.0; Y108 Build/NRD90M) [FBAN/EMA;FBBV/118379517;FBAV/102.0.0.12.163$FB4A$;FBDV/Y108;FBLC/en_US;FBNG/WIFI;FBMNT/METERED;FBDM/{de
SupportsFresco=1 Dalvik/2.1.0 (Linux; U; Android 10; SM-J600G Build/QP1A.190711.020) [FBAN/EMA;FBBV/129523958;FBAV/115.0.0.9.100$FB4A$;FBDV/SM-J600G;FBLC/en_US;FBNG/4G;FBMNT/METI

accountstatus	appid	event_time	bandwidth_usage	is_connect_site	duration	device	proxy_ip_address
0	4.01E+14	1634200336	156	FALSE	10	{"model":"SM-J600G","os":"Android","os_version":"10"}	Redaction - UII
0	4.01E+14	1634200336	156	FALSE	9	{"model":"SM-J600G","os":"Android","os_version":"10"}	
0	4.01E+14	1634200336	156	FALSE	8	{"model":"SM-J600G","os":"Android","os_version":"10"}	
0	4.01E+14	1634217393	156	FALSE	29	{"model":"Y108","os":"Android","os_version":"7.0"}	
0	4.01E+14	1634217393	156	FALSE	11	{"model":"Y108","os":"Android","os_version":"7.0"}	
0	4.01E+14	1634229399	156	FALSE	10	{"model":"Moto E (4)","os":"Android","os_version":"7.1.1"}	
0	4.01E+14	1634229399	156	FALSE	9	{"model":"Moto E (4)","os":"Android","os_version":"7.1.1"}	
0	4.01E+14	1634229399	156	FALSE	10	{"model":"Moto E (4)","os":"Android","os_version":"7.1.1"}	
0	4.01E+14	1634219193	156	FALSE	10	{"model":"Y108","os":"Android","os_version":"7.0"}	
0	4.01E+14	1634215477	156	FALSE	9	{"model":"SM-J600G","os":"Android","os_version":"10"}	

mobile_hashid	method	memory_usage	error_code	bg_ping	rid_for_userid	sdk_type	environment	mobile_client_locale
63e13b3408a94d05b73961dea241347a	gr:post:/graphql	2097152	100	FALSE	Redaction - UII	0	1	en_US
63e13b3408a94d05b73961dea241347a	gr:post:/graphql	2097152	100	FALSE		0	1	en_US
63e13b3408a94d05b73961dea241347a	gr:post:/graphql	2097152	100	FALSE		0	1	en_US
78db186046536500956bb420df2d84b4	gr:post:/graphql	2097152	100	FALSE		0	1	en_US
78db186046536500956bb420df2d84b4	gr:post:/graphql	2097152	100	FALSE		0	1	en_US
ec7cb6eb81022bd6654257f5d807609c	gr:post:/graphql	2097152	100	FALSE		0	1	en_US
ec7cb6eb81022bd6654257f5d807609c	gr:post:/graphql	2097152	100	FALSE		0	1	en_US
ec7cb6eb81022bd6654257f5d807609c	gr:post:/graphql	2097152	100	FALSE		0	1	en_US
78db186046536500956bb420df2d84b4	gr:post:/graphql	2097152	100	FALSE		0	1	en_US
63e13b3408a94d05b73961dea241347a	gr:post:/graphql	2097152	100	FALSE		0	1	en_US

predicted_network_provider_id	predicted_connection_type	is_shadow_req	predicted_carrier_label	brand	client_country_code	predicted_carrier_id	target_id
	mobile			Samsung			
	mobile			Samsung			
	mobile			Samsung			
	mobile						
	mobile						
	wifi			Motorola			
	wifi			Motorola			
	wifi			Motorola			
	mobile						
	mobile			Samsung			

ip_carrier_id	applied_api_version	query_name	preview_mode	dialtone_mode	enabled_features	fbtrace_id	communityid	accountid
2034		MarketplaceJewelCountQuery	FALSE	FALSE	[]	F3AeQ+D52PH		Redaction - UII
2034		GroupsTabBadgeCount	FALSE	FALSE	[]	DgcoWXeUEUq		
2034		UserEventsQuery	FALSE	FALSE	[]	AwDPhNY1a1U		
3078		UserEventsQuery	FALSE	FALSE	[]	BSO0ZYxgVwD		
3078		MarketplaceJewelCountQuery	FALSE	FALSE	[]	H4INd0AbZgv		
0		MarketplaceJewelCountQuery	FALSE	FALSE	[]	C/awX48rZuw		
0		GroupsTabBadgeCount	FALSE	FALSE	[]	FQVAOBgktbs		
0		UserEventsQuery	FALSE	FALSE	[]	EUGra4KHSTa		
3078		UserEventsQuery	FALSE	FALSE	[]	DxHV+ybKchH		
2034		UserEventsQuery	FALSE	FALSE	[]	AX/ga2h9fwt		

ip_address_unsafe	rid_for_accountid	is_developer_site_call	www_request_id	ds	fb_mobile_app	ts
Redaction - UII	Redaction - UII	FALSE	A8pMj9-SLdt9tXbcFKT8OVY	10/14/2021	turducken	2021-10-14+08:00:99
		FALSE	AuqIgx9AazB8B0-DQksvApK	10/14/2021	turducken	2021-10-14+08:00:99
		FALSE	Ay6aLf3i6EZQ7km2WoYs7Dt	10/14/2021	turducken	2021-10-14+08:00:99
		FALSE	A7k327ppeOLRNeld1QLBEqP	10/14/2021	turducken	2021-10-14+13:00:99
		FALSE	AeE4Vkj5V3bOvJxl4BpGbRN	10/14/2021	turducken	2021-10-14+13:00:99
		FALSE	ABnD2ykn8YHYv0dxEQZ_msn	10/14/2021	turducken	2021-10-14+16:00:99
		FALSE	AP6w7XjW8QlBArKn08ePbaK	10/14/2021	turducken	2021-10-14+16:00:99
		FALSE	Aku7VrxMRXN2Gqaz1EA2oro	10/14/2021	turducken	2021-10-14+16:00:99
		FALSE	AsJgbOczjNcixAs9t7uezz5	10/14/2021	turducken	2021-10-14+13:00:99
		FALSE	AGmLu9NT22_96WFGxvqWX51	10/14/2021	turducken	2021-10-14+12:00:99

Exhibit C

to Ji Declaration

Highly Confidential

api_hits_www

Column	Type	Description
ds	string	Date on which the data were generated
ts	string	Time stamp showing hour on which the data were generated
event_time	double	The webserver timestamp at the moment that log() was called.
bandwidth_usage	bigint	Number of bytes in the response
is_connect_site	boolean	Whether the connection came from Facebook Connect
duration	bigint	Response time for the API call
session_origin	bigint	The install flow from which the session originated
proxy_ip_address	string	RemoteAccess::isProxied, if exists
userid	bigint	The result of calling getUserID on a supplied ViewerContext. This could be a person's ID, a page's, oran application's. If you only want the person behind the page or app, you want the 'accountid' field.
method	string	The API method that was logged.
memory_usage	bigint	Max amt of memory used during the request
rid_for_userid	bigint	The Clementine RID for this UserID, if exists.
impersonating_appid	bigint	App_id that was being impersonated
bg_ping	boolean	Was the request made while app was backgrounded?
error_code	bigint	Error code thrown
ip_address	string	The IP address from which the user actually accessed Facebook.
sdk_type	bigint	Which Facebook SDK was used to make this call
environment	bigint	The TFBEnv value.
method_type	bigint	Category of the API method
user_agent	string	The UserAgent field from a user's browser
accountstatus	bigint	The result of calling getConfirmedValue on the EntAccount inside the ViewerContext
appid	bigint	Application ID
target_id	string	Graph API ID of the root node of the request
applied_api_version	string	API version that was effectively applied to this request
fbtrace_id	string	Fbtrace ID, unique for each request
communityid	bigint	Workplace Community ID
is_workplace_app	boolean	Workplace App
accountid	bigint	The logged in user ACCOUNT's ID (different from UserID, which could be a page)
ip_address_unsafe	string	The IP address from which the user claims to have accessed Facebook.
rid_for_accountid	bigint	The Clementine RID for this UserID, if exists
is_developer_site_call	boolean	Did the request originate from developers facebook.com
www_request_id	string	Request ID, unique for each request
install_id	bigint	Install ID
actor	string	Represents the currently logged-in user. It is derived from the ViewerContext on www

Exhibit D

to Ji Declaration

Highly Confidential

api_hits_www Sample

event_time	bandwidth_usage	is_connect_site	duration	session_origin	proxy_ip_address	userid	method	memory_usage	rid_for_userid
1634278364	81	FALSE	5	0	Redaction - UII	Redaction - UII	gr:post:/logging_client_events	2097152	Redaction - UII
1634272501	107	TRUE	2	0			gr:post:/logging_client_events	2097152	
1634273977	107	TRUE	4	0			gr:post:/logging_client_events	2097152	
1634277545	107	TRUE	3	0			gr:post:/logging_client_events	2097152	
1634279889	1905	FALSE	8	0			gr:get:Application/mobile_sdk_gk	2097152	
1634278218	107	TRUE	2	0			gr:post:/logging_client_events	2097152	
1634278604	81	FALSE	2	0			gr:post:/logging_client_events	2097152	
1634278363	1059	FALSE	8	0			gr:get:Application	2097152	
1634278364	858	FALSE	10	0			gr:get:Application	2097152	
1634278364	81	FALSE	5	0			gr:post:/logging_client_events	2097152	

impersonating_appid	bg_ping	error_code	ip_address	sdk_type	environment	method_type
0	FALSE	0	Redaction - UII	0	1	2
0	FALSE	0		407	1	2
0	FALSE	0		409	1	2
0	FALSE	0		0	1	2
0	FALSE	0		167837700	1	1
0	FALSE	0		408	1	2
0	FALSE	0		0	1	2
0	FALSE	0		155910276	1	1
0	FALSE	0		167837700	1	1
0	TRUE	0		0	1	2

user_agent
Instagram 210.0.0.0.46 Android (29/10; 440dpi; 1080x2132; Xiaomi/POCO; M2010J19CI; citrus; qcom; en_IN; 324158833)
Instagram 207.0.0.28.118 (iPhone12,5; iOS 14_7_1; en_US; en-US; scale=3.00; 1242x2688; 320361397) AppleWebKit/420+
Instagram 209.1.0.25.113 (iPhone12,1; iOS 14_6; en_EG; ar-EG; scale=2.00; 828x1792; 324511477) AppleWebKit/420+
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36
FBAndroidSDK.8.1.0
Instagram 208.0.0.26.131 (iPhone11,6; iOS 14_8; en_US; en-US; scale=3.00; 1242x2688; 322184766) AppleWebKit/420+
Instagram 208.0.0.32.135 Android (30/11; 440dpi; 1080x2254; Xiaomi/Redmi; Redmi Note 9 Pro; joyeuse; qcom; tr_TR; 323102559)
FBAndroidSDK.5.11.2
FBAndroidSDK.8.1.0
[FBAN/MQTT;FBAV/73.0.14;FBBV/309825994;FBDM/{density=3.0,width=1080,height=2110};FBLC/ru_RU;FBCR/MegaFon;FBMF/samsung;FBBD/samsung;FBPN/com.facebook.services;FBDV/SM-A405FM

accountstatus	appid	target_id	applied_api_version	fbtrace_id	communityid	is_workplace_app	accountid	ip_address_unsafe	rid_for_accountid	is_developer_site_call
	5.67E+14		v1.0			FALSE	Redaction - UII	Redaction - UII	Redaction - UII	FALSE
	1.24E+14		v1.0			FALSE				FALSE
	1.24E+14		v1.0			FALSE				FALSE
	9.37E+14		v4.0			FALSE				FALSE
	0	5.84E+14	v8.0			FALSE				FALSE
	1.24E+14		v1.0			FALSE				FALSE
	5.67E+14		v1.0			FALSE				FALSE
	0	1.64E+14	v5.0			FALSE				FALSE
	0	1.44E+15	v8.0			FALSE				FALSE
l;FBSV/11;FBLR/(	7.25E+14		v2.0			FALSE				FALSE

www_request_id	install_id	actor	ds	ts
		Redaction - UII	10/14/2021	2021-10-15+06:00:99
			10/14/2021	2021-10-15+04:00:99
			10/14/2021	2021-10-15+04:00:99
			10/14/2021	2021-10-15+05:00:99
			10/14/2021	2021-10-15+06:00:99
			10/14/2021	2021-10-15+06:00:99
			10/14/2021	2021-10-15+06:00:99
			10/14/2021	2021-10-15+06:00:99
			10/14/2021	2021-10-15+06:00:99
			10/14/2021	2021-10-15+06:00:99

Exhibit E

to Ji Declaration

Highly Confidential

api_hits_app_method_r

Column	Type	Description
ds	string	Date DS string
appid	string	App ID
method	string	API method used
cnt	bigint	Number of API hits
users	bigint	Approximate number of distinct SIDs with API hits
cnt_successful	bigint	Number of API hits, excluding errors

Exhibit F

to Ji Declaration

Highly Confidential

api_hits_app_method_r Sample

appid	method	cnt	users	cnt_successful	ds
1.69276E+15	gr:get:Application/mobile_sdk_gk	659	385	659	10/14/2021
5.72656E+14	gr:get:User/accounts	2	1	2	10/14/2021
3.06288E+14	gr:get:InstagramOembed	112	1	0	10/14/2021
1.57565E+15	unknown	1	1	0	10/14/2021
7.59809E+14	gr:batch	4	1	4	10/14/2021
3.16271E+14	auth.androidauthorizeapp	11	6	5	10/14/2021
2.58889E+14	gr:get:URL	4	1	4	10/14/2021
2.2654E+15	gr:post:Page/feed	3	2	3	10/14/2021
1.3887E+14	gr:batch	21	1	21	10/14/2021
8.88371E+14	gr:get:ShadowIGUser/insights	2	1	2	10/14/2021